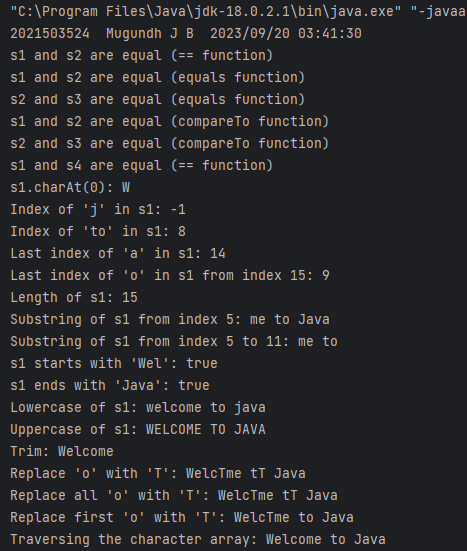
|  |
| --- |
| **Ex.No: 7 DATE: 20-09-23 STRING HANDLING** |

1) Aim: To write a java program to perform string methods by considering the given string inputs String s1=”Welcome to Java”; String s2=s1; String s3=new String(“Welcome to Java”); String s4=s1.intern();

Code:

import java.time.LocalDateTime;  
import java.time.format.DateTimeFormatter;  
  
public class stringMethods3524 {  
 public static void main(String[] args) {  
 // Code for getting the current date and time  
 DateTimeFormatter dtf = DateTimeFormatter.*ofPattern*("yyyy/MM/dd HH:mm:ss");  
 LocalDateTime now = LocalDateTime.*now*();  
 System.*out*.println("2021503524 " + "Mugundh J B " + dtf.format(now));  
  
 // Initialize some strings  
 String s1 = "Welcome to Java";  
 String s2 = s1;  
 String s3 = new String("Welcome to Java");  
 String s4 = s1.intern();  
  
 // String methods  
 if (s1 == s2) System.*out*.println("s1 and s2 are equal (== function)");  
 if (s2 == s3) System.*out*.println("s2 and s3 are equal (== function)");  
 if (s1.equals(s2)) System.*out*.println("s1 and s2 are equal (equals function)");  
 if (s2.equals(s3)) System.*out*.println("s2 and s3 are equal (equals function)");  
 if (s1.compareTo(s2) == 0) System.*out*.println("s1 and s2 are equal (compareTo function)");  
 if (s2.compareTo(s3) == 0) System.*out*.println("s2 and s3 are equal (compareTo function)");  
 if (s1 == s4) System.*out*.println("s1 and s4 are equal (== function)");  
  
 // String methods and operations  
 System.*out*.println("s1.charAt(0): " + s1.charAt(0));  
 System.*out*.println("Index of 'j' in s1: " + s1.indexOf('j'));  
 System.*out*.println("Index of 'to' in s1: " + s1.indexOf("to"));  
 System.*out*.println("Last index of 'a' in s1: " + s1.lastIndexOf('a'));  
 System.*out*.println("Last index of 'o' in s1 from index 15: " + s1.lastIndexOf("o", 15));  
 System.*out*.println("Length of s1: " + s1.length());  
 System.*out*.println("Substring of s1 from index 5: " + s1.substring(5));  
 System.*out*.println("Substring of s1 from index 5 to 11: " + s1.substring(5, 11));  
 System.*out*.println("s1 starts with 'Wel': " + s1.startsWith("Wel"));  
 System.*out*.println("s1 ends with 'Java': " + s1.endsWith("Java"));  
 System.*out*.println("Lowercase of s1: " + s1.toLowerCase());  
 System.*out*.println("Uppercase of s1: " + s1.toUpperCase());  
 System.*out*.println("Trim: " + " Welcome ".trim());  
 System.*out*.println("Replace 'o' with 'T': " + s1.replace('o', 'T'));  
 System.*out*.println("Replace all 'o' with 'T': " + s1.replaceAll("o", "T"));  
 System.*out*.println("Replace first 'o' with 'T': " + s1.replaceFirst("o", "T"));  
  
 // Traverse the character array within the string  
 System.*out*.print("Traversing the character array: ");  
 for (char ch : s1.toCharArray()) {  
 System.*out*.print(ch);  
 }  
 }  
}

Output:



1.2) Write a program to show that String is immutable in java.

Code:

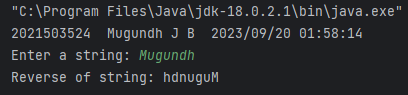
import java.time.LocalDateTime;  
import java.time.format.DateTimeFormatter;  
public class stringImmutable3524 {  
 public static void main(String[] args) {  
 // Code for displaying student information and current date/time  
 DateTimeFormatter dtf = DateTimeFormatter.*ofPattern*("yyyy/MM/dd HH:mm:ss");  
 LocalDateTime now = LocalDateTime.*now*();  
 System.*out*.println("2021503524 " + "Mugundh J B " + dtf.format(now));  
  
 // Create a string object  
 String originalString="Hello, ";  
  
 // Concatenate another string  
 originalString += "World!"; // Modifying the original string  
  
 System.*out*.println("Strings are immutable in Java:\n");  
 //Check if it is the same reference  
 if (originalString == "Hello, World!") {  
 System.*out*.println("The strings are the same reference.");  
 } else {  
 System.*out*.println("The strings are different references.");  
 }  
  
 //Check if the content is the same  
 if (originalString.equals("Hello, World!")) {  
 System.*out*.println("The content of the strings is the same.");  
 } else {  
 System.*out*.println("The content of the strings is different.");  
 }  
 }  
}

2) Write a java program to read the string and displays the reverse of the string

Code:

import java.time.LocalDateTime;  
import java.time.format.DateTimeFormatter;  
import java.util.Scanner;  
  
public class stringReverse3524 {  
  
 // Function to perform string reverse operation  
 public static void reverse(String str){  
 // Since, string is immutable, converting string to char array for mutations  
 char[] s = str.toCharArray();  
  
 int n = str.length(), mid = n / 2;  
 // Performing swap until mid  
 for(int i = 0; i <= mid; i++){  
 // Swapping the characters  
 char t = s[i];  
 s[i] = s[n - i - 1];  
 s[n - i - 1] = t;  
 }  
  
 // Converting char array to string  
 String reverse = new String(s);  
  
 System.*out*.println("Reverse of string: " + reverse);  
 }  
  
 public static void main(String[] args) {  
 // Code for displaying student information and current date/time  
 DateTimeFormatter dtf = DateTimeFormatter.*ofPattern*("yyyy/MM/dd HH:mm:ss");  
 LocalDateTime now = LocalDateTime.*now*();  
 System.*out*.println("2021503524 " + "Mugundh J B " + dtf.format(now));  
  
 Scanner in = new Scanner(System.*in*);  
 // Getting string as input  
 System.*out*.print("Enter a string: ");  
 String str = in.nextLine();  
  
 *reverse*(str); // Function to reverse the string  
  
 }  
}

Output:

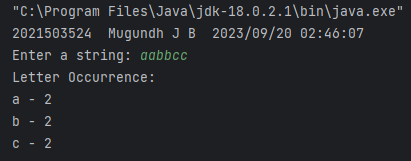


3) Write a java program to count the number of occurrence of the each letter in the given string

Code:

import java.time.LocalDateTime;  
import java.time.format.DateTimeFormatter;  
import java.util.Scanner;  
  
public class letterOccurance3524 {  
 public static void main(String[] args) {  
 // Code for displaying student information and current date/time  
 DateTimeFormatter dtf = DateTimeFormatter.*ofPattern*("yyyy/MM/dd HH:mm:ss");  
 LocalDateTime now = LocalDateTime.*now*();  
 System.*out*.println("2021503524 " + "Mugundh J B " + dtf.format(now));  
  
 Scanner in = new Scanner(System.*in*);  
  
 System.*out*.print("Enter a string: ");  
 String s = in.nextLine();  
  
 int[] freq = new int[256];  
 int n = s.length();  
  
 for (int i = 0; i < n; i++)  
 freq[s.charAt(i)] += 1; // Use the character itself as the index  
  
 System.*out*.println("Letter Occurrence: ");  
 for (int i = 0; i < 256; i++) {  
 // Converting int to ASCII char code  
 char ch = (char) i;  
 if (freq[i] > 0)  
 System.*out*.printf("%c - %d\n", ch, freq[i]); // Print the character 'ch'  
 }  
 }  
}

Output:

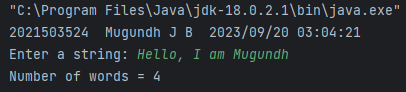


4) Write a Java program to count the number of words in the given string

Code:

import java.time.LocalDateTime;  
import java.time.format.DateTimeFormatter;  
import java.util.Scanner;  
  
public class countWords3524 {  
 public static void main(String[] args) {  
 // Code for displaying student information and current date/time  
 DateTimeFormatter dtf = DateTimeFormatter.*ofPattern*("yyyy/MM/dd HH:mm:ss");  
 LocalDateTime now = LocalDateTime.*now*();  
 System.*out*.println("2021503524 " + "Mugundh J B " + dtf.format(now));  
  
 Scanner in = new Scanner(System.*in*);  
  
 //Declaring the string variable and obtaining user input  
 System.*out*.print("Enter a string: ");  
 String str = in.nextLine();  
  
 //Initialize the variable to track the number of words  
 int count = 0;  
 boolean inWord = false;  
  
 //For loop to traverse the string  
 for (char c: str.toCharArray())  
 {  
 if (Character.*isWhitespace*(c)) {  
 // If a space is encountered, it indicates the end of a word  
 inWord = false;  
 } else {  
 // If a non-space character is encountered  
 // and we are not already in a word, it indicates the start of a new word  
 if (!inWord) {  
 count++;  
 inWord = true;  
 }  
 }  
 }  
  
 System.*out*.println("Number of words = " + count);  
 }  
}

Output:

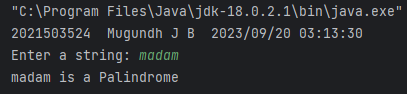


5.1) Write a java program to check the given string is palindrome or not (Example:Race car)

Code:

import java.time.LocalDateTime;  
import java.time.format.DateTimeFormatter;  
import java.util.Scanner;  
  
public class strPalindrome3524 {  
 // Function to check if a given string is a palindrome  
 public static boolean isPalindrome(String str) {  
 int left = 0, right = str.length() - 1;  
 while (left < right) {  
 if (str.charAt(left++) != str.charAt(right--))  
 return false; // If characters at left and right positions do not match, it's not a palindrome  
 }  
 return true; // If the loop completes without returning false, it's a palindrome  
 }  
  
 public static void main(String[] args) {  
 // Code for displaying student information and current date/time  
 DateTimeFormatter dtf = DateTimeFormatter.*ofPattern*("yyyy/MM/dd HH:mm:ss");  
 LocalDateTime now = LocalDateTime.*now*();  
 System.*out*.println("2021503524 " + "Mugundh J B " + dtf.format(now));  
  
 Scanner in = new Scanner(System.*in*);  
  
 System.*out*.print("Enter a string: ");  
 String s = in.nextLine();  
  
 // Check if the entered string is a palindrome and display the result  
 if (*isPalindrome*(s))  
 System.*out*.println(s + " is a Palindrome");  
 else  
 System.*out*.println(s + " is not a Palindrome");  
 }  
}

Output:

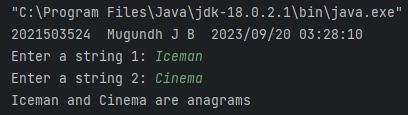


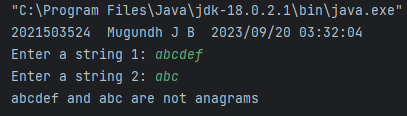
5.2) Write a java program to check the given string is anagram or not (Example Iceman vs Cinema)

Code:

import java.time.LocalDateTime;  
import java.time.format.DateTimeFormatter;  
import java.util.Locale;  
import java.util.Scanner;  
  
public class strAnagram3524 {  
 // Function to check if two strings are anagrams  
 public static void checkAnagrams(String str1, String str2) {  
 // Initialize an array to store letter frequencies for lowercase English letters  
 int[] freq = new int[26];  
  
 // Convert both input strings to lowercase for case-insensitive comparison  
 String s1 = str1.toLowerCase(), s2 = str2.toLowerCase();  
  
 // Count the frequency of each letter in the first string  
 for (int i = 0; i < s1.length(); i++)  
 freq[s1.charAt(i) - 'a'] += 1;  
  
 int flag = 0;  
 // Compare the frequency of letters in the second string  
 for (int i = 0; i < s2.length(); i++) {  
 if (--freq[s2.charAt(i) - 'a'] < 0) {  
 System.*out*.println(str1 + " and " + str2 + " are not anagrams");  
 flag = 1;  
 break;  
 }  
 }  
 if(flag == 0)  
 System.*out*.println(str1 + " and " + str2 + " are anagrams");  
 }  
  
 public static void main(String[] args) {  
 // Code for displaying student information and current date/time  
 DateTimeFormatter dtf = DateTimeFormatter.*ofPattern*("yyyy/MM/dd HH:mm:ss");  
 LocalDateTime now = LocalDateTime.*now*();  
 System.*out*.println("2021503524 " + "Mugundh J B " + dtf.format(now));  
  
 Scanner in = new Scanner(System.*in*);  
  
 System.*out*.print("Enter a string 1: ");  
 String str1 = in.nextLine();  
 System.*out*.print("Enter a string 2: ");  
 String str2 = in.nextLine();  
  
 // Check if the lengths of the two strings are different  
 if (str1.length() != str2.length())  
 System.*out*.println(str1 + " and " + str2 + " are not anagrams");  
 else  
 *checkAnagrams*(str1, str2); // Call the function to check if the strings are anagrams  
 }  
}

Output:



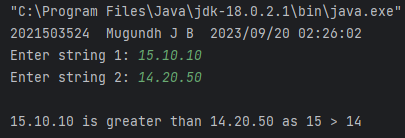


6) Write a java program that read a two string of the given format and compares the string

Code:

import java.time.LocalDateTime;  
import java.time.format.DateTimeFormatter;  
import java.util.Scanner;  
  
public class productVersion3524 {  
 public static void main(String[] args) {  
 // Code for displaying student information and current date/time  
 DateTimeFormatter dtf = DateTimeFormatter.*ofPattern*("yyyy/MM/dd HH:mm:ss");  
 LocalDateTime now = LocalDateTime.*now*();  
 System.*out*.println("2021503524 " + "Mugundh J B " + dtf.format(now));  
  
 Scanner in = new Scanner(System.*in*);  
  
 System.*out*.print("\nEnter string 1: ");  
 String s1 = in.nextLine();  
  
 System.*out*.print("Enter string 2: ");  
 String s2 = in.nextLine();  
  
 // Split the input strings by dot (.) to compare version numbers  
 String[] res1 = s1.split("\\.");  
 String[] res2 = s2.split("\\.");  
  
 // Compare version numbers component by component  
 for(int i = 0; i < res1.length; i++) {  
 // Accessing the substrings of version numbers and parsing them as integer  
 if(Integer.*parseInt*(res1[i]) > Integer.*parseInt*(res2[i])) {  
 System.*out*.println(s1 + " is greater than " + s2 + " as " + res1[i] + " > " + res2[i]);  
 break;  
 }  
 else if(Integer.*parseInt*(res1[i]) < Integer.*parseInt*(res2[i])) {  
 System.*out*.println(s2 + " is greater than " + s1 + " as " + res2[i] + " > " + res1[i]);  
 break;  
 }  
 }  
 }  
}

Output:

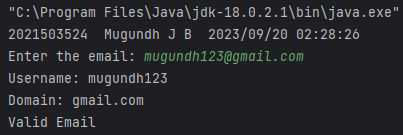


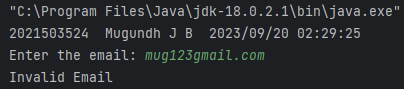
7) Write a java program using String methodsto compare the email is valid is invalid and returnsthe username and domain name i) Valid Username: numbers[0-7], alphabets[a-z][A-Z], underscore, dot, hypen and plus characters ii) Presence of @ symbol iii) Presence of domainname.com or .in or .edu Hint use givenstring.split(“@”) to find specific user(case-insensitive:jc\_vp) and specific domain (case-insensitive: gmail.com) for example [jc\_vp@gmail.com](mailto:jc_vp@gmail.com).

Code:

import java.time.LocalDateTime;  
import java.time.format.DateTimeFormatter;  
import java.util.Scanner;  
  
public class emailValidation3524 {  
 public static boolean isValidEmail(String email) {  
 // Check for the presence of "@" symbol for a valid email  
 if (!email.contains("@")) {  
 return false;  
 }  
  
 // Split the email into username and domain  
 String[] parts = email.split("@");  
 if (parts.length != 2) {  
 return false;  
 }  
  
 // Check the validity of the username  
 String username = parts[0];  
 if (!*isValidUsername*(username)) {  
 return false;  
 }  
  
 // Check the validity of the domain  
 String domain = parts[1].toLowerCase();  
 if (!(domain.endsWith(".com") || domain.endsWith(".in") || domain.endsWith(".edu"))) {  
 return false;  
 }  
  
 // Check for a valid domain name  
 if (!*isValidDomain*(domain)) {  
 return false;  
 }  
  
 return true;  
 }  
  
 public static boolean isValidUsername(String username) {  
 // Valid username: numbers[0-9], alphabets[a-zA-Z], underscore, dot, hyphen, and plus characters  
 String regex = "^[0-9a-zA-Z\_.+-]+$";  
 return username.matches(regex);  
 }  
  
 public static boolean isValidDomain(String domain) {  
 // Valid domain name: Only alphabets and hyphens are allowed  
 String regex = "^[a-zA-Z-.]+$";  
 return domain.matches(regex);  
 }  
  
 public static void main(String[] args) {  
 // Code for getting the current date and time  
 DateTimeFormatter dtf = DateTimeFormatter.*ofPattern*("yyyy/MM/dd HH:mm:ss");  
 LocalDateTime now = LocalDateTime.*now*();  
 System.*out*.println("2021503524 " + "Mugundh J B " + dtf.format(now));  
  
 Scanner in = new Scanner(System.*in*);  
 System.*out*.print("Enter the email: ");  
 String email = in.nextLine();  
  
 if (*isValidEmail*(email)) {  
 String[] parts = email.split("@");  
 String username = parts[0];  
 String domain = parts[1];  
 System.*out*.println("Username: " + username);  
 System.*out*.println("Domain: " + domain);  
 System.*out*.println("Valid Email");  
 } else {  
 System.*out*.println("Invalid Email");  
 }  
  
 }  
}

Output:





8) Write a java program to create a dictionary using 2D string array for any 10 programming languages. Write a method that return the definition for the input of Programming Language name.

Code:

import java.time.LocalDateTime;  
import java.time.format.DateTimeFormatter;  
import java.util.Scanner;  
  
public class progLangDictionary3524 {  
  
 // Method to search for a programming language definition  
 public static String search(String[][] pL, String str) {  
 // Checking the match for programming language in dictionary  
 for (int i = 0; i < 10; i++) {  
 if (str.equalsIgnoreCase(pL[i][0])) // Comparing strings to find out if they are equal, ignoring case differences  
 return pL[i][1];  
 }  
 return "";  
 }  
  
 public static void main(String[] args) {  
 // Code for displaying student information and current date/time  
 DateTimeFormatter dtf = DateTimeFormatter.*ofPattern*("yyyy/MM/dd HH:mm:ss");  
 LocalDateTime now = LocalDateTime.*now*();  
 System.*out*.println("2021503524 " + "Mugundh J B " + dtf.format(now));  
  
 Scanner in = new Scanner(System.*in*);  
  
 // Create a 2D array to store programming language definitions  
 String[][] pL = {  
 {"Java", "Java is a pure object-oriented programming language developed by James Gosling."},  
 {"C++", "C++ is an object-oriented programming language developed by Bjarne Stroustrup."},  
 {"Python", "Python is a high-level programming language known for its readability and versatility."},  
 {"JavaScript", "JavaScript is a versatile scripting language commonly used for web development."},  
 {"C#", "C# is a modern, object-oriented programming language developed by Microsoft."},  
 {"Ruby", "Ruby is a dynamic and expressive scripting language."},  
 {"PHP", "PHP is a server-side scripting language widely used for web development."},  
 {"Swift", "Swift is a fast and powerful programming language developed by Apple for iOS and macOS."},  
 {"Go", "Go is a statically typed, compiled language known for its efficiency and simplicity."},  
 {"Rust", "Rust is a systems programming language focused on safety and performance."}  
 };  
  
 System.*out*.print("Enter string to be searched in dictionary: ");  
 String str = in.nextLine();  
  
 // Search for the programming language definition  
 String res = *search*(pL, str);  
  
 // Display the result or a message if not found  
 if (res != "")  
 System.*out*.println(str + " : " + res);  
 else  
 System.*out*.println("Programming Language not found in Dictionary");  
 }  
}

Output:

